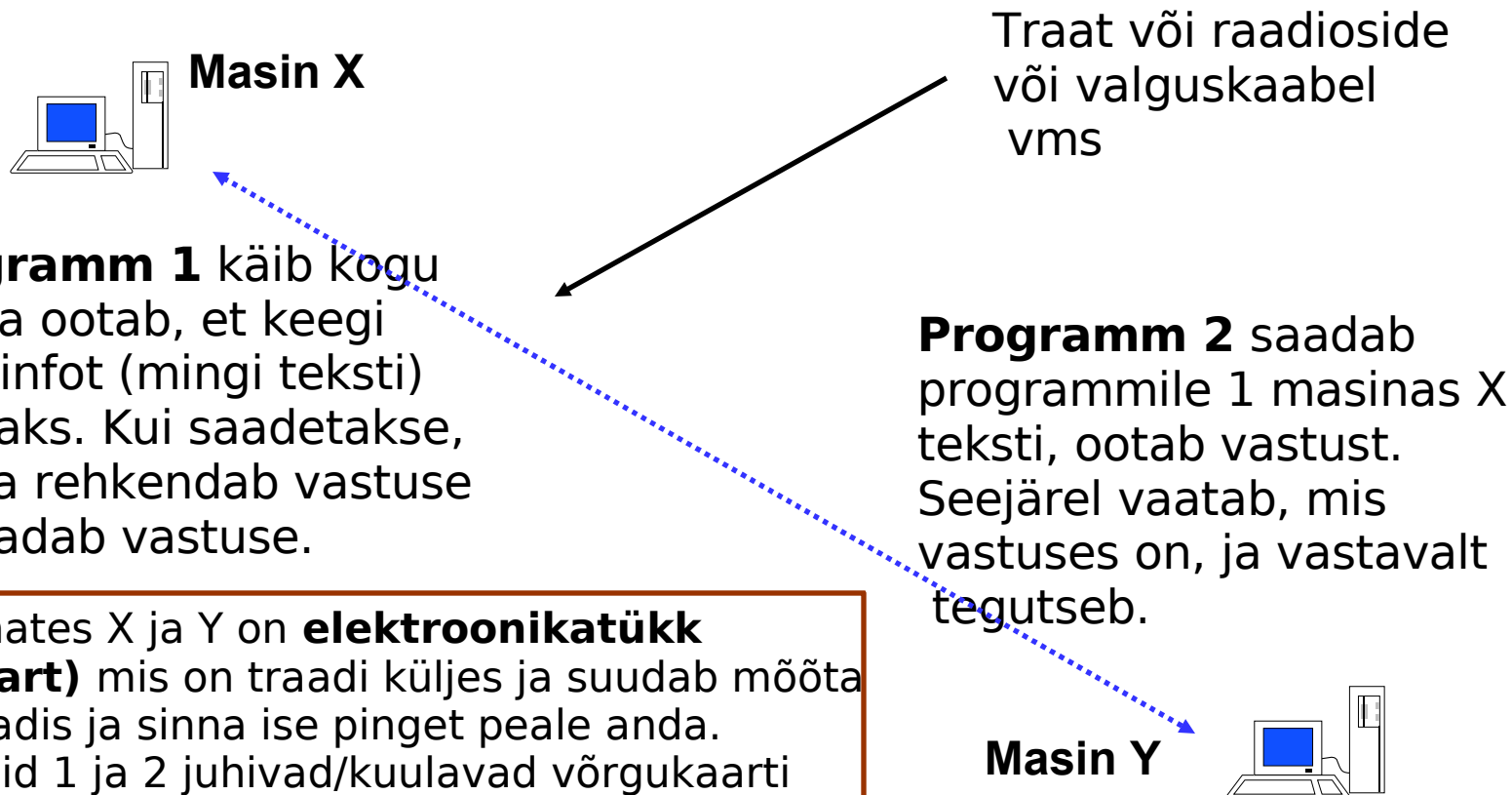

Sissejuhatus infotehnoloogiasse

Loengu ülevaade. Interneti funktsioneerimine

- Kiirülevaade võrgunduse funktsioneerimisest
- Kiirülevaade interneti funktsioneerimisest
- Baasmehhanismi, IP funktsioneerimine
- TCP ja UDP: kaks täiendavat protokollu IP-le
- Üldpilt kihilisest protokollindusest: ISO/OSI mudel
- Muud sideprotokollid
- HTTP protokoll
- DNS funktsioneerimine
- Näiteid brauseriga ja telnetiga

Kiirülevaade võrgunduse funktsioneerimisest

■ Üldpilt:

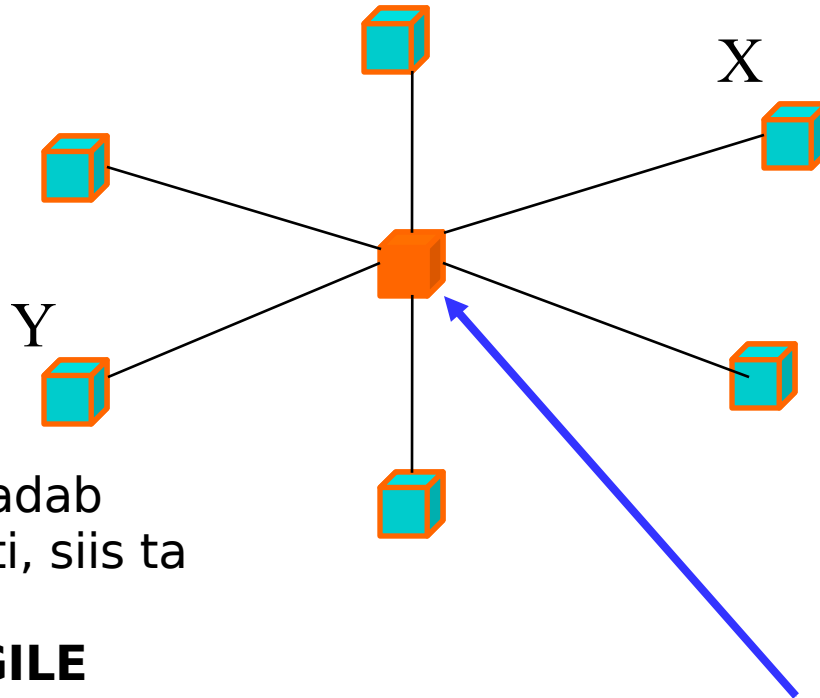


NB! Masinates X ja Y on **elektroonikatükk (võrgukaart)** mis on traadi küljes ja suudab mööta pinget traadis ja sinna ise pinget peale anda. Programmid 1 ja 2 juhivad/kuulavad võrgukaarti kaardi draiveriprogrammi abil.

Interneti funktsioneerimine 1

- Üldpilt sama, mis eelmisel slaidil.
- Küsimused, mis tahavad vastust:
 - Kui arvuti Y on traadiga ühenduses mitme eri arvutiga (X, Z, U, ...) siis kuidas saata nimelt soovitud arvutisse (meie näites X) oma info?
 - Kui info hulk on väga suur (näiteks 700 MB), kas saaks seda üksikute väiksemate tükkidena saata?
 - Kuidas kontrollida, kas tükid jõudsid pärale?
 - Mis teha, kui mõni tükk kaduma läheb?
 - Kuidas arvuti X tükid õieti kokku oskab panna?

Hulk arvuteid võrgus traatipidi koos: ethernet



Kui arvuti Y saadab arvutile X teksti, siis ta saadab selle traatipidi **KÕIGILE** arvutitele võrgus, aga teksti juures on öeldud, et **AINULT** X (MAC aadress!) peab seda kuulama. **Teised ignoreerivad.**

Igale arvutil on oma **unikaalne NIMI** ehk aadress

Ethernetis:

MAC aadress:
48 bitine arv

HUB on lihtsalt traatide **“harupesa”**, ühendab kõik kokku

NB! Korraga on traadil ainult üks teade!

Ethernet Frame

62 bits	Preamble used for bit synchronization
2 bits	Start of Frame Delimiter
48 bits	Destination Ethernet Address
48 bits	Source Ethernet Address
16 bits	Length or Type
46 -1500 bytes	Data
32 bits	Frame Check Sequence

Hulk arvuteid võrgus traatipidi koos: internet

**Teksti küljes on
alati IP
aadress.**

Iga võrgus olev
masin suunab
saadud teksti
õiges suunas.
Kuidas?

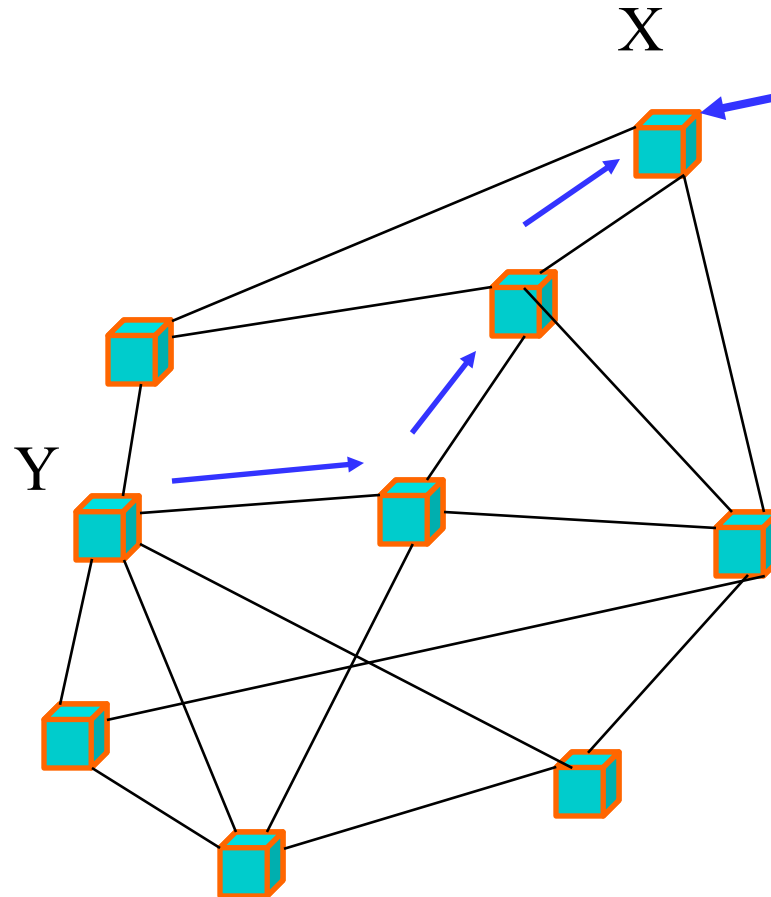
Nn “routing
table”
ütleb, vaadates
IP aadressi.

Igas arvutis on nn
“**router**”-

programm, mis
kuulab traati ja
saadab saadud

teksti õiges
suunas edasi.

Osa arvuteid võrgus on “lihtsalt” harilikud
arvutid, osa spetsiaalsed “võrgu-ruuterid”



Igal arvutil
on oma
**unikaalne
NIMI**
ehk aadress

Internetis:
IP aadress:
32 bitine arv

32 bitti on 4
baiti. IP aadress
kirjutatakse
enamasti
“inimloetavalt”
nii (näide):
193.40.254.179

Interneti alusprotokoll (IP) kasutamise põhi-ideed

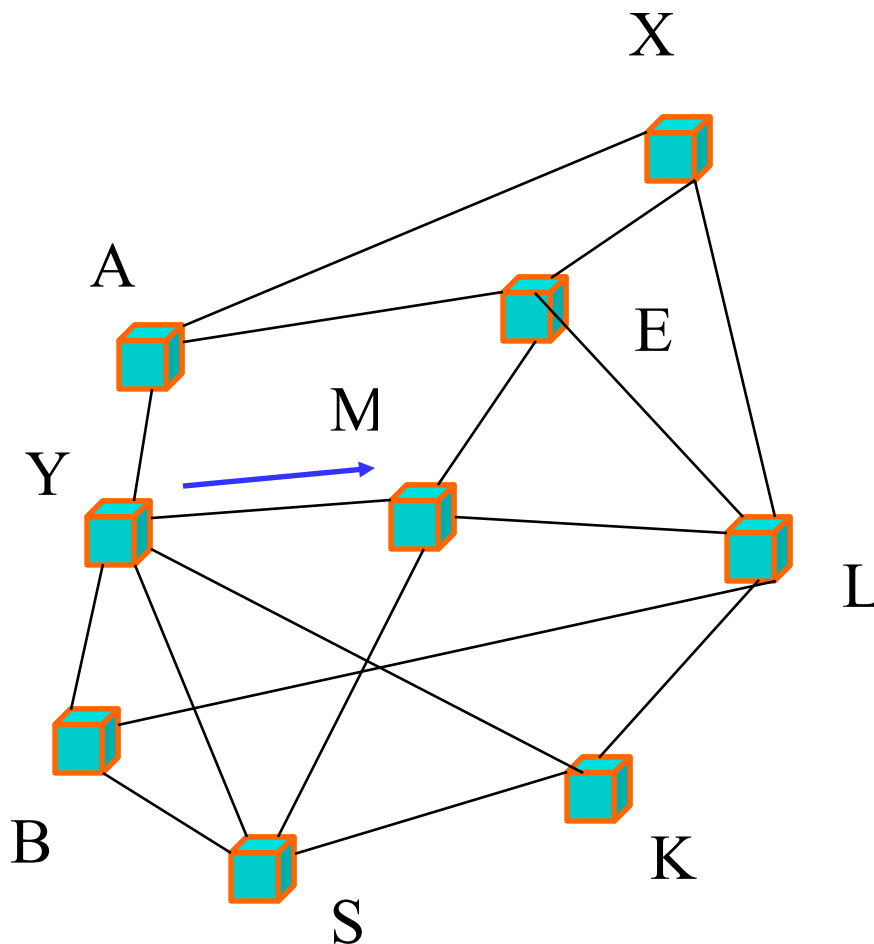
- **Aadress:** iga saadetava tekstijupiga pannakse kaasa selle masina aadress, kuhu tekst saata tuleb.
- See on analoogiline hariliku ümbrikuga:
 - tekst on ümbriku sees.
 - ümbriku peal on saaja aadress (ja tagaküljel ka saatja aadress)
 - 4 baiti IP aadressis on nagu aadressi-read: maja, tänav, linn, riik
 - ruuter-arvutid on nagu postkontorid ja sorteerimispunktid
 - traadid on nagu veomasinad, mis posti kontorite vahel veavad
- **Paketid:** harilikku ümbrikku aga ei mahu terve entsoklüpeedia kümme köidet! Mida teha? Paneme eraldi köited eri ümbrikutesse ja saadame kümme ümbrikku.
- Nii ka internetis:
 - Pikad tekstid lõhutakse väiksemateks juppideks (nn pakettideks)
 - Paketid saadetakse eraldi, igalühel aadress peal.

Kuidas info siis liigub? 1

Y tahab saata teksti “Ahoi, tere” masinale IP-aadressiga X.

Ruuterprogramm otsustab, kumbat juhet (tegelikult, vahemasina aadressi) valida.

Otsutagu näiteks masina M kasuks.

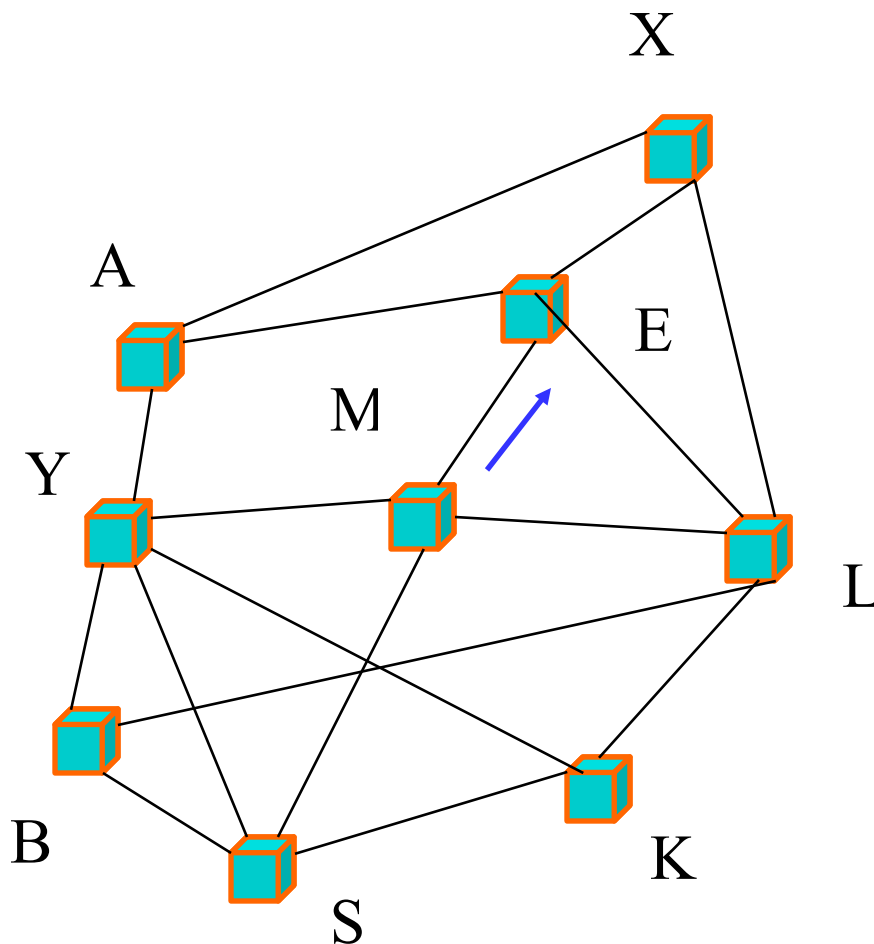


Kuidas info siis liigub? 2

M peab saatma teksti “Ahoi, tere” edasi masinale IP-aadressiga X.

Ruuterprogramm otsustab, kumbat juhet (tegelikult, vahemasina aadressi) valida.

Otsutagu näiteks masina E kasuks.

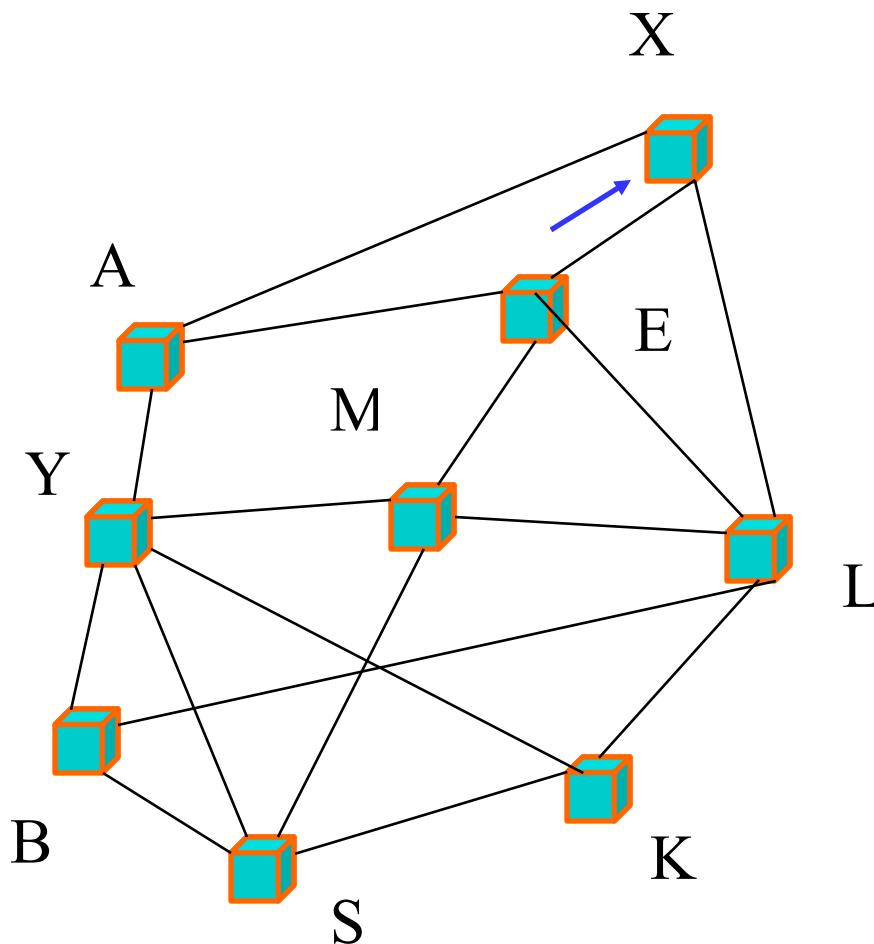


Kuidas info siis liigub? 3

E peab saatma teksti “Ahoi, tere” edasi masinale IP-aadressiga X.

Ruuterprogramm otsustab, kumbat juhet (tegelikult, vahemasina aadressi) valida.

Otsutagu (ilmselt!) masina X kasuks.

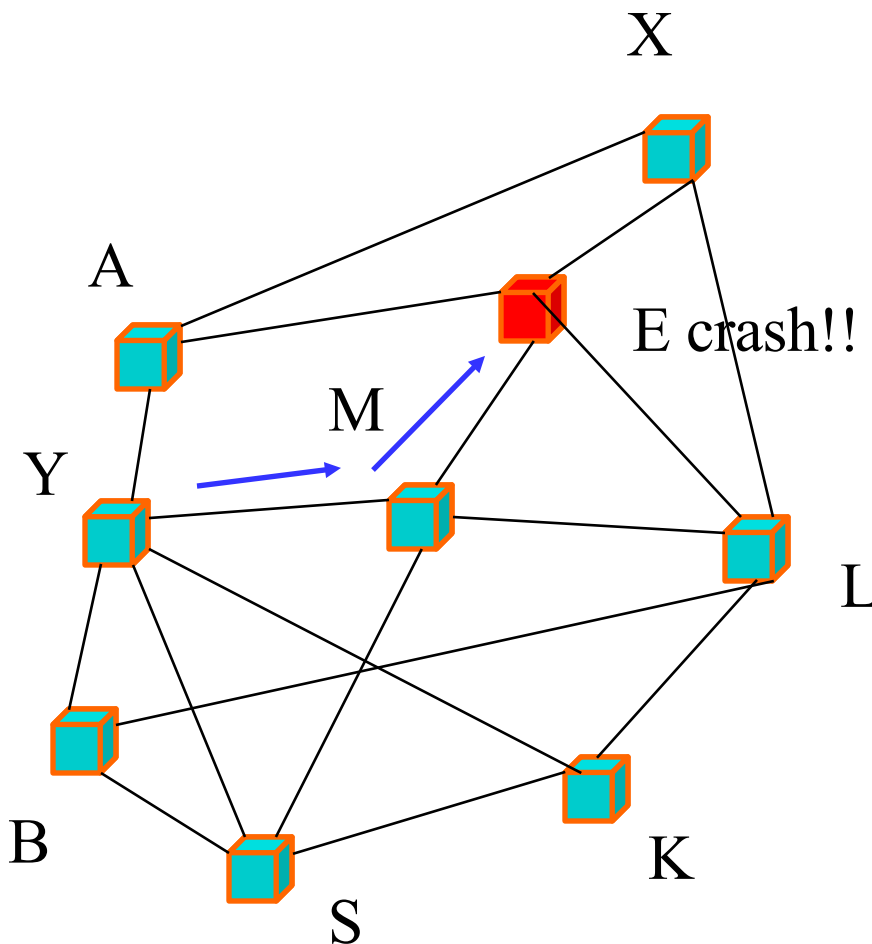


Paketid võivad kaduma minna

Y peab saatma
teksti “Ahoi, tere”
edasi masinale
IP-aadressiga X.

Teade läheb
masinani E, aga
siis E crashib
(läheb katki,
programm läheb
segamini, vool
läheb ära vms).

**Võrgu
alusprotokoll:
IP ise ei
kontrolli, kas
teade
jõudis pärale!**



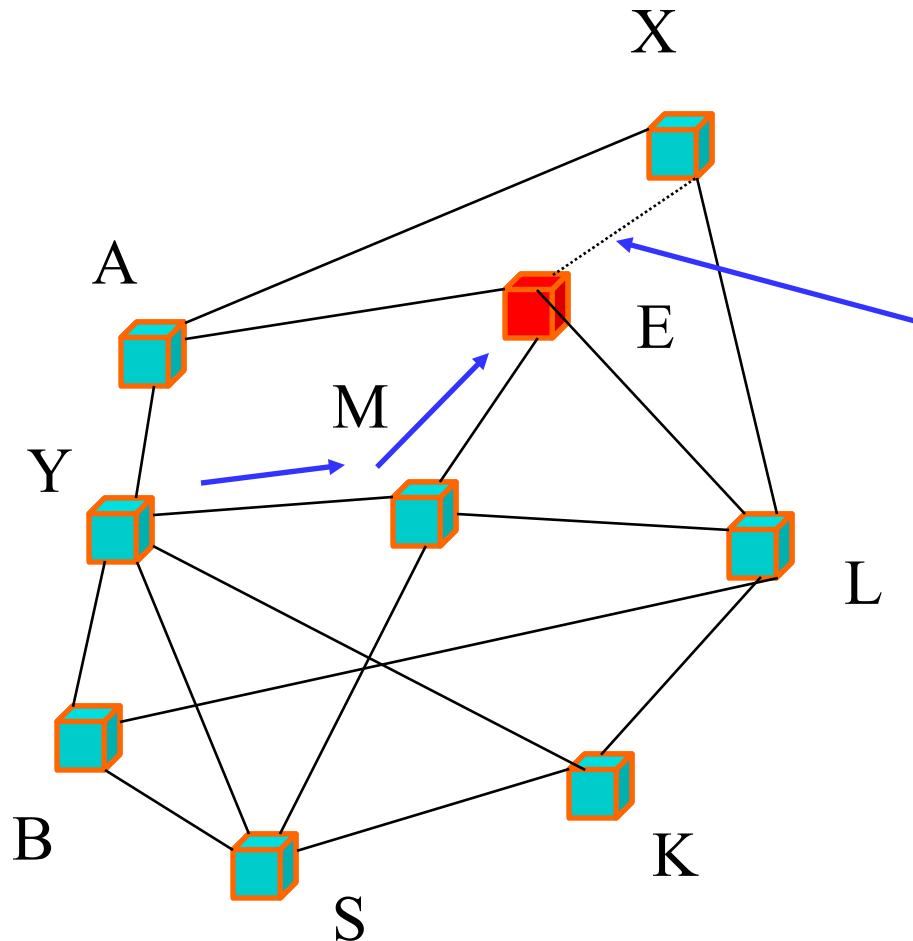
Paketid võivad minna eri teid pidi, eri kiirusega

Y peab saatma teksti "**Ahoi, tere**" edasi masinale IP-aadressiga X.

A lõhub pika teksti kaheks paketiks!

Osa "**Ahoi,**" läheb otse E->X, aga siis läheb E-X otseühendus rikki.

E leiab, et saab saata teate teise osa "**tere**" X-ile ka masina L kaudu.

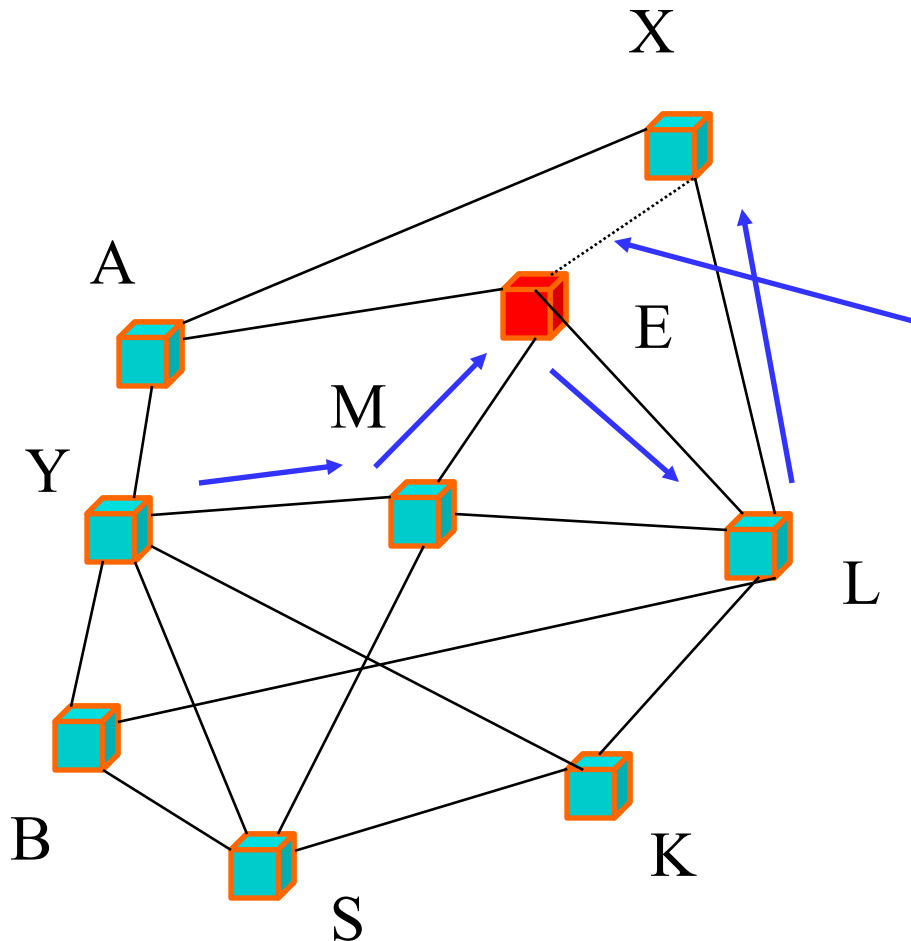


Ühendus
puruneb!

Paketid võivad minna eri teid pidi, eri kiirusega

E suunabki teate masinale L, kes suunab selle masinasse X.

Kaks teksti osa liikusid eri teid pidi!

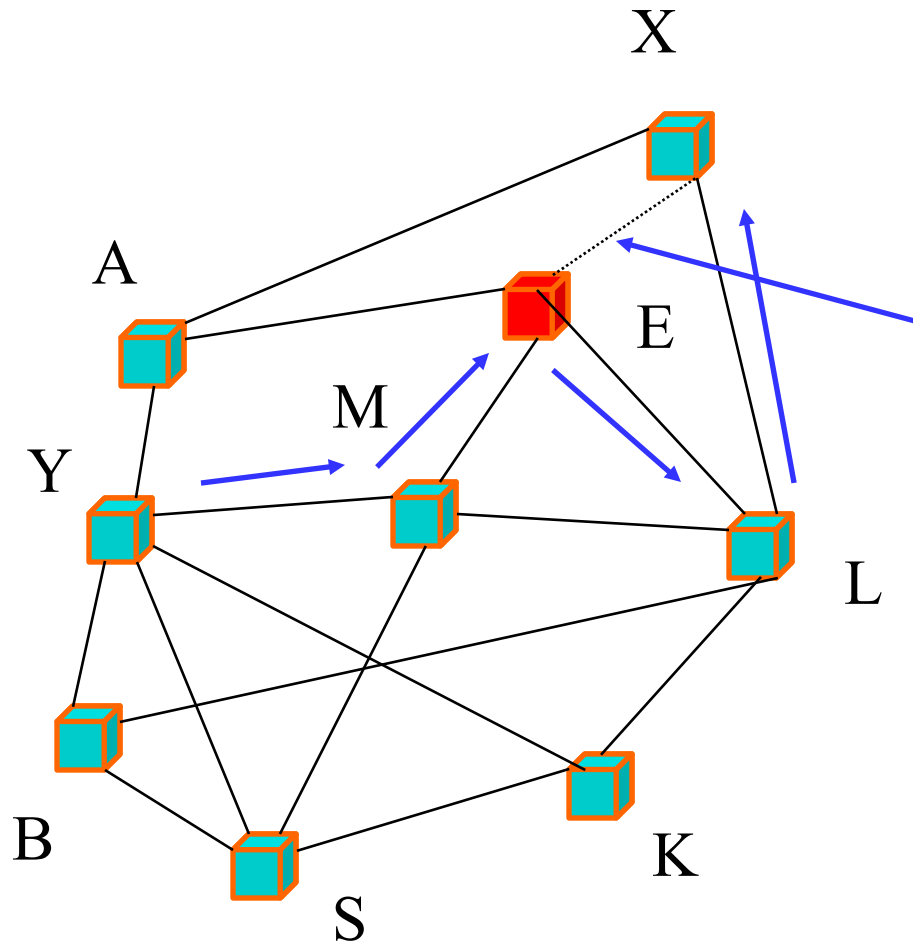


Ühendus purunenud, aga pole hullu ...

Paketid võivad minna eri teid pidi, eri kiirusega

Hiljem saadab
Y uue teate
"Tere taas"
masinale X.

E suunab
esimese osa
"Tere" läbi
masina L.



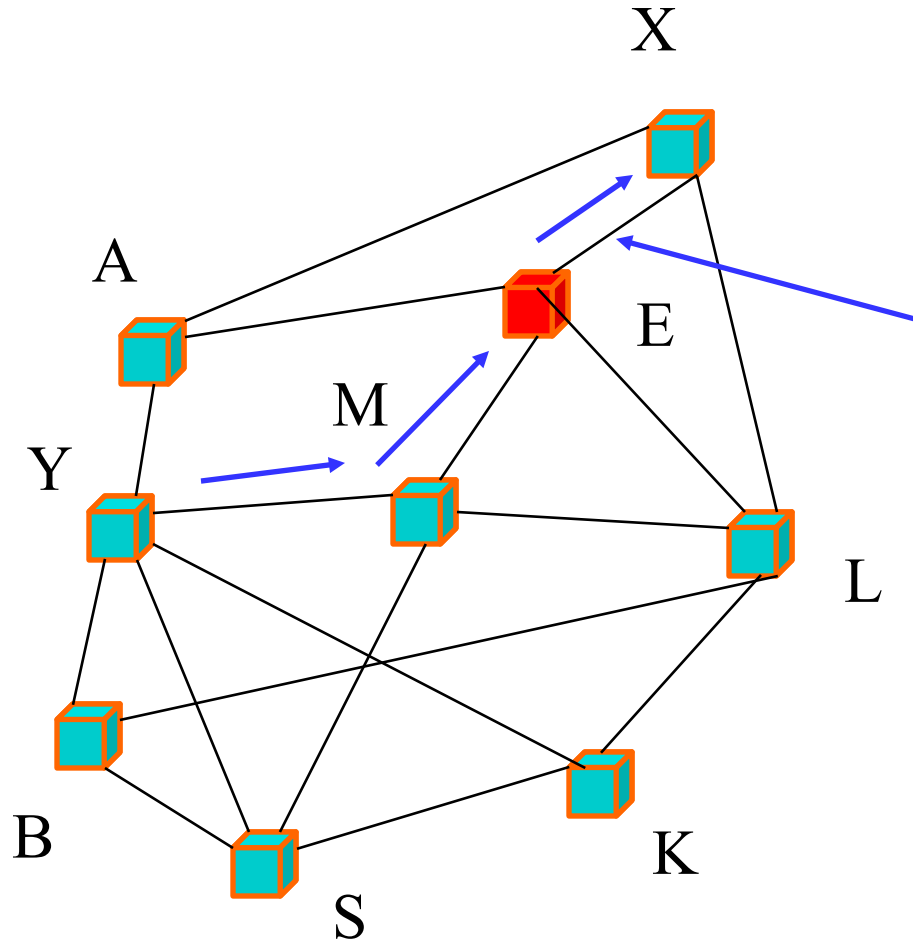
Ühendus
rikkis

Paketid võivad minna eri teid pidi, eri kiirusega

....

Siis läheb
otseühendus
korda,
ja E suunab osa
“taas” otse
masinale X.

**Teine osa võib
seega jõuda
kohale enne
esimest!**



Ühendus
läheb korda!

Interneti alusprotokoll: IP (internet protocol)

- IP protokoll on **kokkulepe**, et kuidas infot saata ja sellest aru saada tuleb.
- IP protokoll lubab saata ainult väikeseid tekstijuppe.
- Iga tekstijupi ette pannakse lisainfo (päis ehk header), mis ütleb, et:
 - kuhu see tekst siis saata tuleb (IP aadress)
 - kust tekst tuli (saatja IP aadress)
 - Hulga lisainfot ka veel



IP (Internet Protocol)

- **TCP/IP pere tööhobune**
- **RFC 791**
- **Garanteerib marsruutimise, st minemise õiges suunas**
- **Mitteusaldusväärne - ei taga kohalejõudmist**
 - Kui sisendpuhver on täis, siis ignoreerib
- **Ei loo kanalit**
 - iga datagrammi käsitletakse sõltumatult

Interneti alusprotokoll: IP (internet protocol)

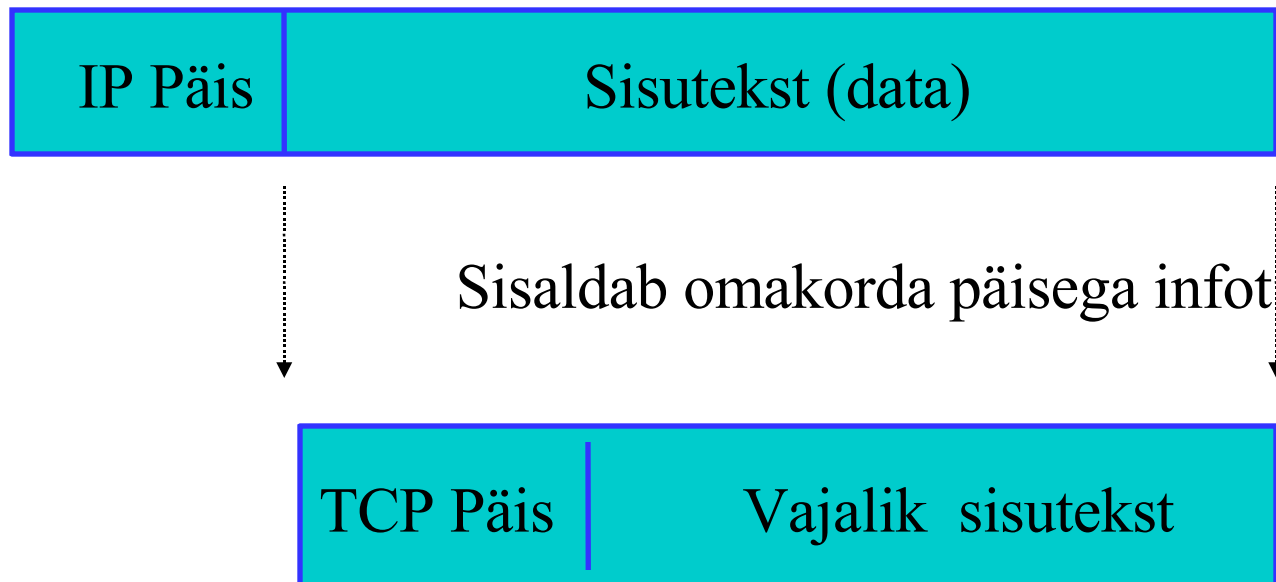
- Konkreetsemalt (pole vaja pähe jätta!)

INTERNET DATAGRAM FORMAT

0	4	8	16	24	31
VERS	HLEN	SERVICE TYPE	TOTAL LENGTH		
IDENTIFICATION			FLAGS	FRAGMENT OFFSET	
TIME TO LIVE		PROTOCOL	HEADER CHECKSUM		
SOURCE INTERNET ADDRESS					
DESTINATION INTERNET ADDRESS					
OPTIONS (IF ANY)				PADDING	
DATA					
DATA					

Interneti järgmised protokollid: UDP ja TCP

- Kaks põhi-protokolli, mis kasutavad IP-d.
 - **UDP** (user datagram protocol). Ei kontrollita, kas info jõudis päralt.
 - **TCP** (transfer control protocol). Toimub kontroll.
- Kumbalgi puhul (TCP näitel):



UDP (User Datagram Protocol)

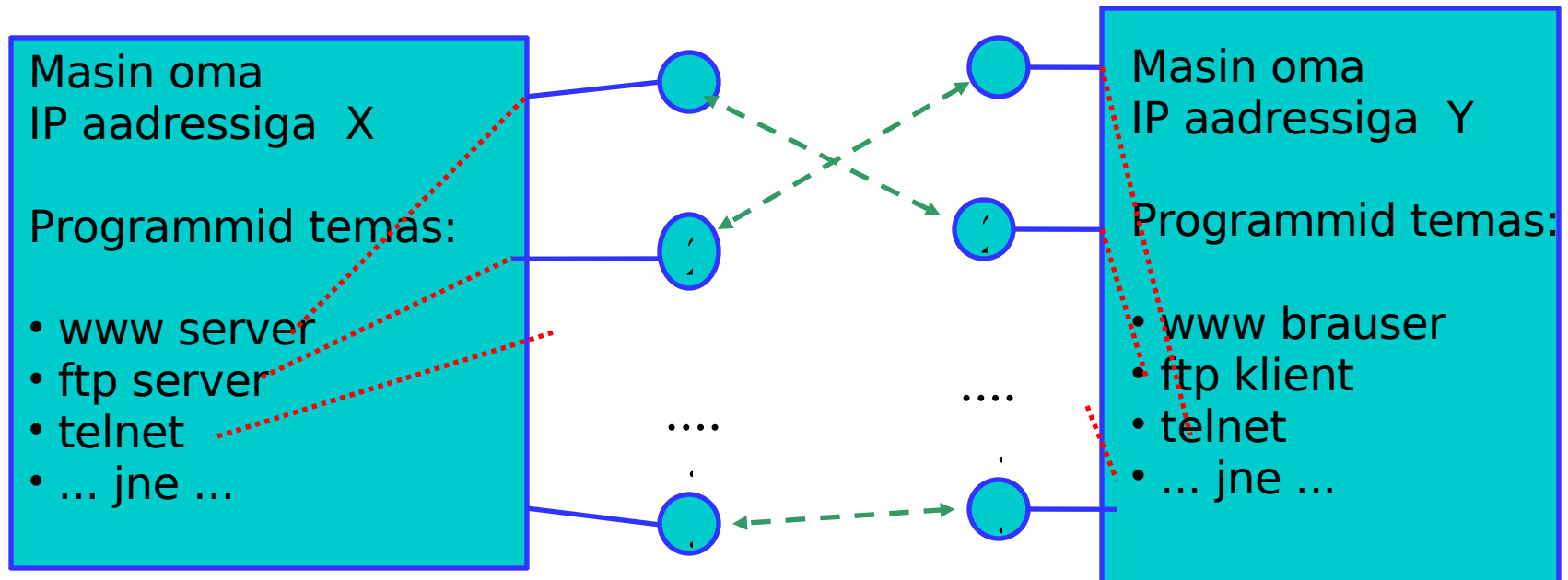
- Iga rakenduse väljund tekitab uue datagrammi
- Ei taga usaldatavust
- Datagrammi ehitus:
 - lähte- ja sihtport (kumbki kaks baiti)
 - datagrammi pikkus (kaks baiti)
 - kontrollsumma (kaks baiti, pole kohustuslik)
 - andmeosa (varieeruva pikkusega hulk baite)
- **UDP paketi maksimaalpikkus on seega 64 kilobaiti.**
- Kontrollsumma vea puhul unustatakse datagramm
- Rakendused: DNS, NFS, TFTP

TCP(Transmission Control Protocol)

- Ühendusorienteeritud
- Usaldatav
- Voo tüüpi
 - Jagab voo segmentideks
 - Saates käivitab taimeri ja ootab kinnitust
 - Kinnitab saadud segmendid
 - Kontrollsumma päisest ja andmetest
 - Korrastab segmentide järjestuse
 - Unustab dublikaadid
 - Kontrollib voo mahtu

Pordid

- Masinas elab korraka palju programme ja igaüks tahab saada/saata oma pakette. Millisele programmile konkreetne pakett saata?
- Programmidele antakse kasutamiseks **nummerdatud TCP või UDP port (need ei ole füüsilised pistikud!)**
- Ühenduse määravad: IP aadressid, pordid, protokoll



Pordid

■ Portide liigitus:

- 1 - 1023 üldtuntud pordid, sh
 - 21 FTP – TCP
 - 23 telnet – TCP
 - 517 UDP
- 1024 – 5000 klientprogrammide ajutised pordid
- >5000 muud (mitte-üldtuntud) serverid

Klient - server mudel

■ Üldine algoritm

- server ootab mingil üldteada pordil ühenduskutseid
- klient reserveerib dünaamilise pordi
- klient saadab serverile ühenduskutse (koos oma pordinumbriga)
- server vastab kliendile tema pordinumbril

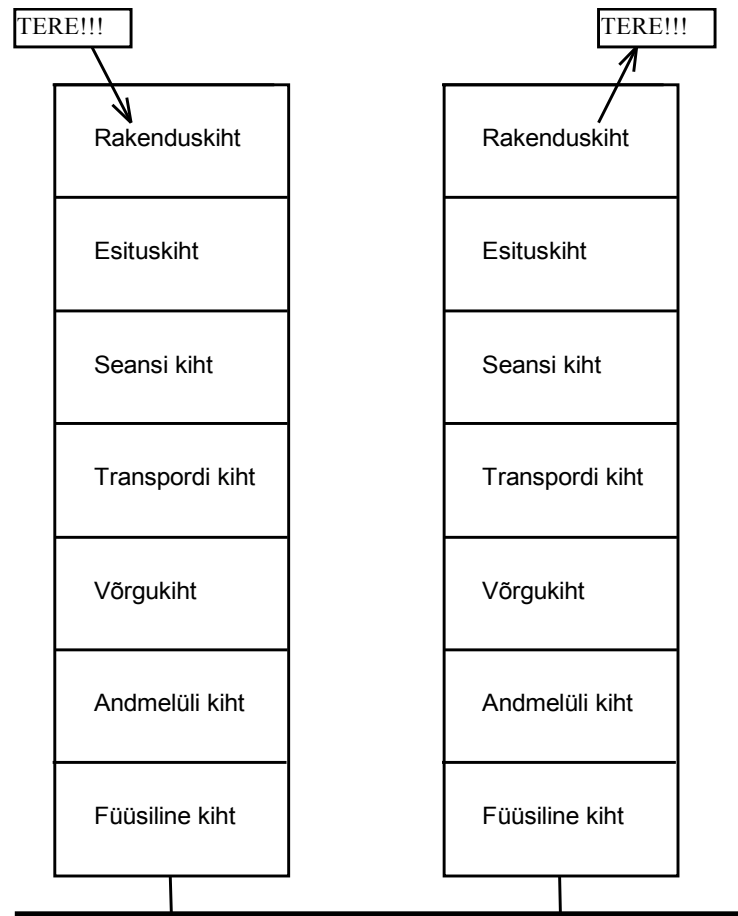
■ Itereeriv server

- Ei võta vastu uut kutset enne eelneva töötlemist

■ Paralleelne server

- Käivitab iga kutse jaoks eraldi serveri ajutise pordinumbriga selle ühenduse jaoks.
- Seega on mitme ajas lähestikuse kutse järel masinas käimas mitu “koopiat” ehk protsessi või threadi ühest serverist
- Kasutajatele paistab, et need “koopiad” käivad korraga, tegelikult hoolitseb selle “kooskäimise simulatsiooni” eest opsüsteem.

ISO - OSI mudel



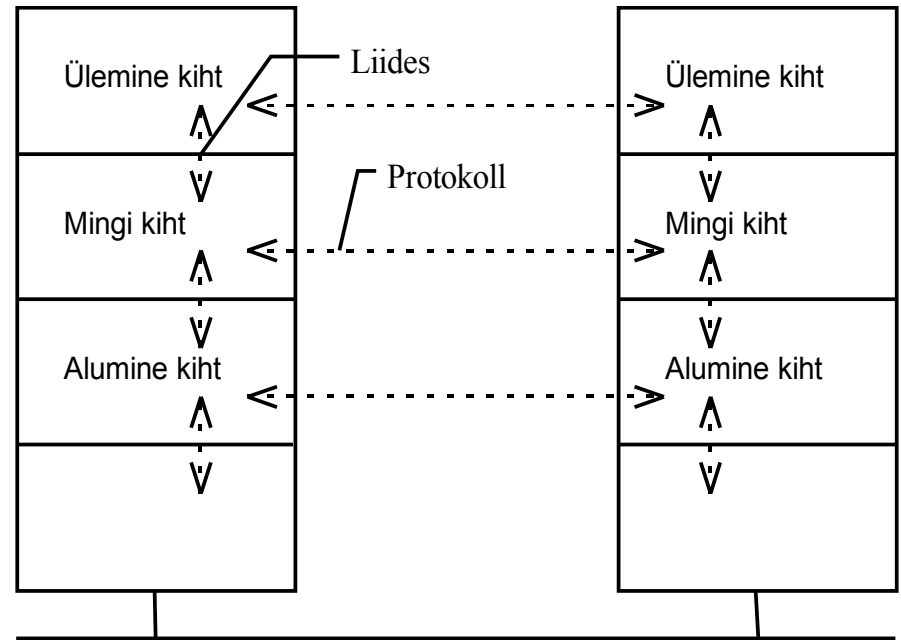
OSI mudeli kihid

■ Liides (interface)

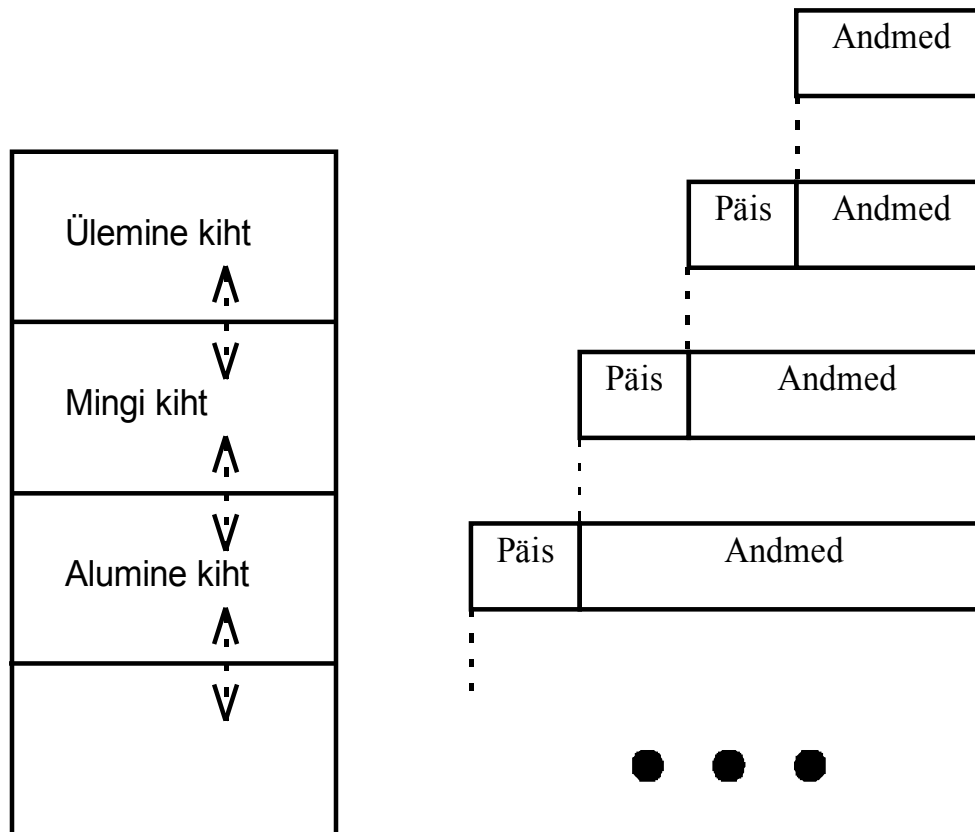
Sama süsteemi eri
kihtide suhtlusviis
omavahel

■ Protokoll (protocol)

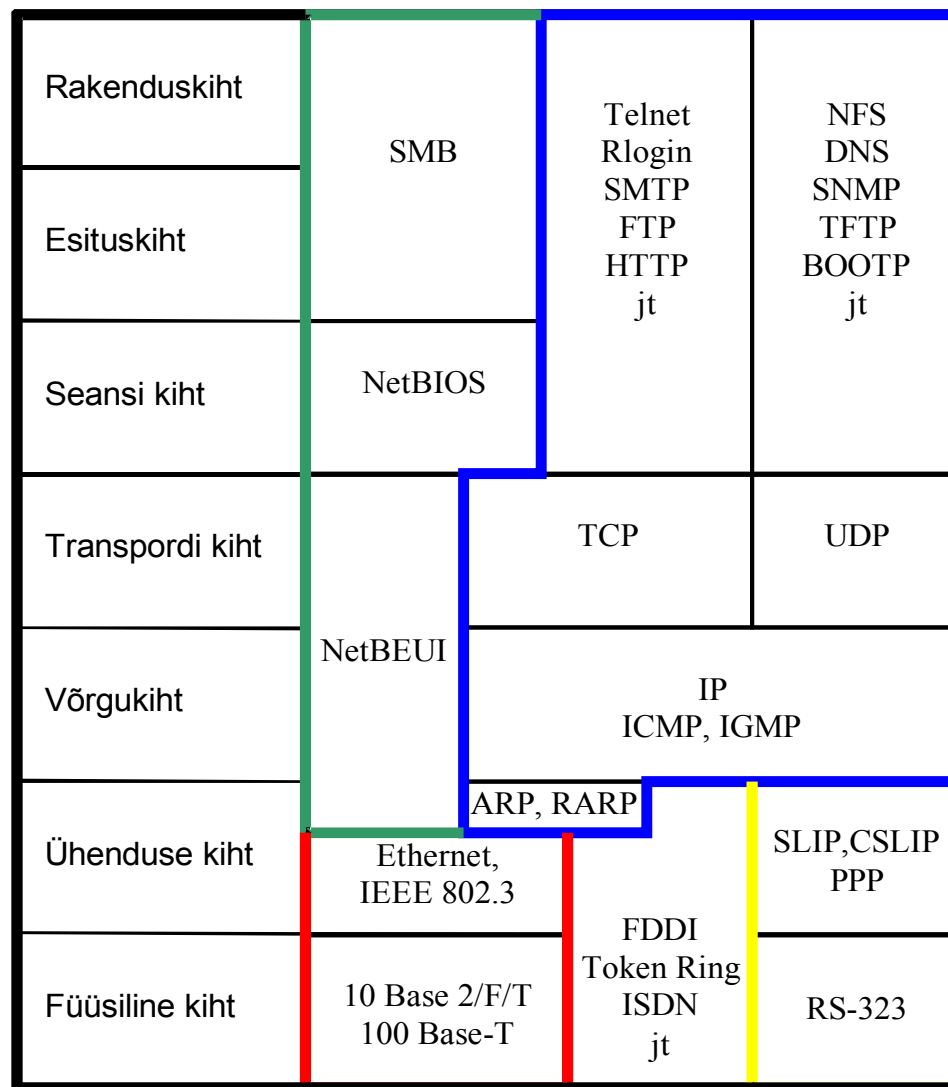
Eri süsteemide samade
kihtide suhtlusviis
omavahel



Kapseldamine

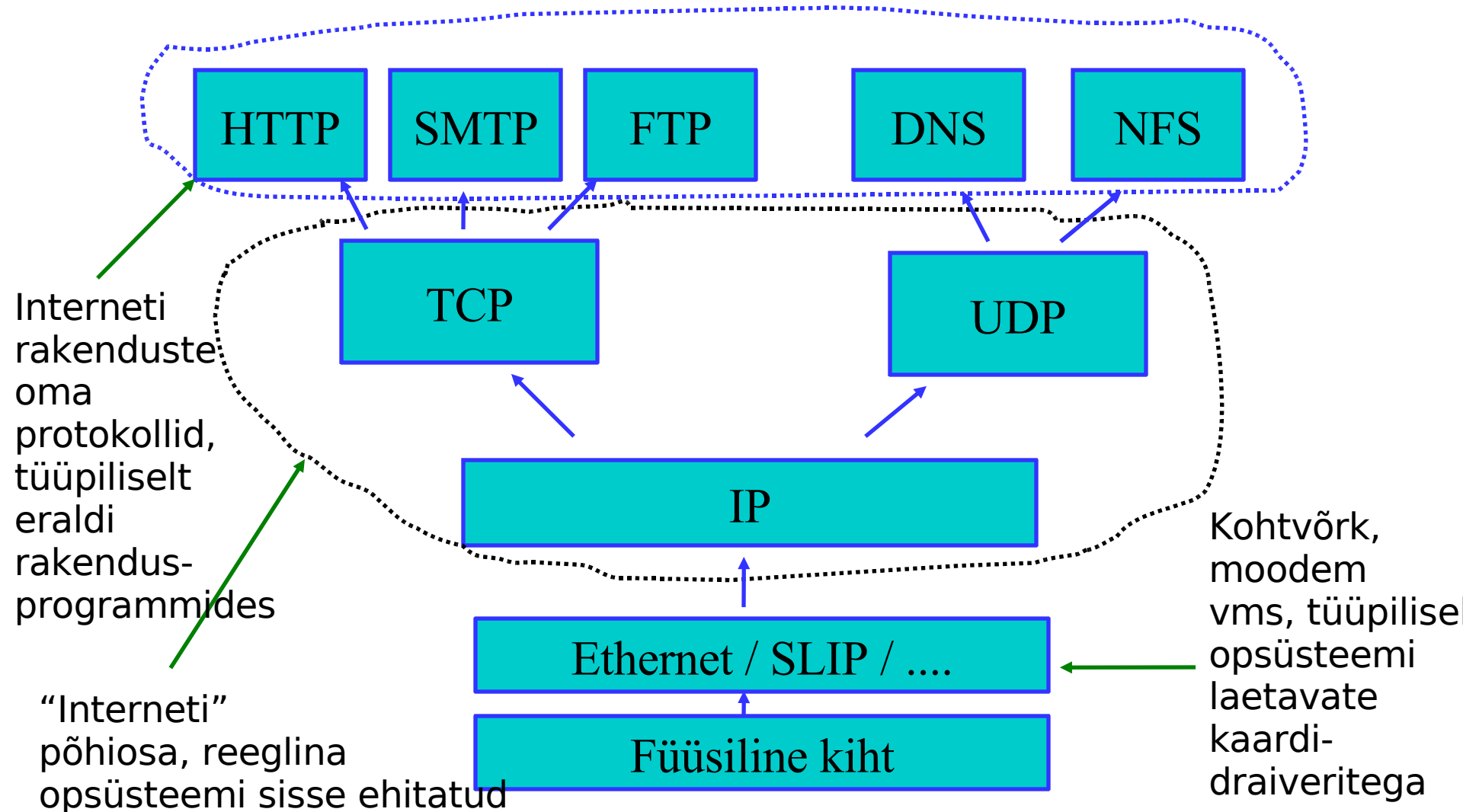


Protokollipered (internet ja mõned muud)



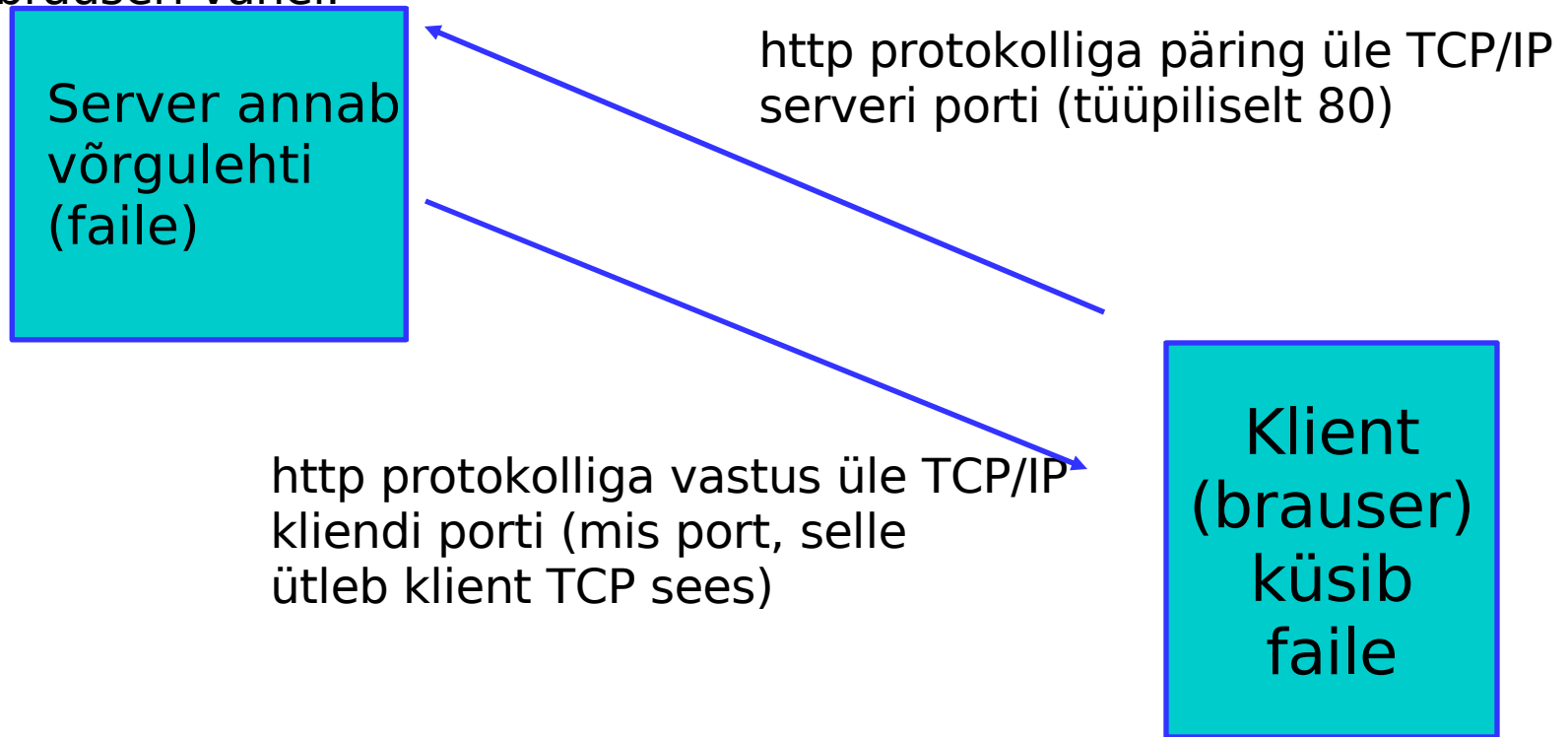
Sinine
osa on
interneti
protokollid

Vahekokkuvõte protokollindusest internetis



HTTP ühendused: failide küsimine ja nende andmine

- **HTTP on omaette protokoll**, mida kasutatakse veebilehtede, piltide, tekstifailide, zip failide jne jne saatmiseks veebiserveri ja brauseri vahel.



Milline on HTTP protokollis päring?

- **HTTP päring on sisuliselt tekstiline käsk serverile:** “anna mulle selline fail”, kus näidatakse ära:
 - konkreetne küsimus-käsk
 - faili asukoht ja nimi
 - protokoll, mida küsija kasutab
 - ja soovi korral lisainfot, nagu küsija programmi tüüp (Mozilla, le, Konqueror, ..)
- **Näiteks:**
 - Võtame telnet ühenduse masina `deepthought.ttu.ee` porti 80
 - Tipime sisse, seejärel tipime kaks reavahetust:

```
GET /it/index.html HTTP/1.0
```


Milline on HTTP protokollis vastus?

- HTTP ei ole ehitatud “biti või baidi” tasemel, vaid teksti ridade kaupa: päis, tühi rida, tekstiread.

päis

```
HTTP/1.1 200 OK
Date: Thu, 06 Nov 2003 13:50:07 GMT
Server: Apache/1.3.19 (Unix) PHP/4.1.1
Last-Modified: Sat, 10 Apr 1999 09:29:18 GMT
ETag: "46d8-297-370f19ee"
Accept-Ranges: bytes
Content-Length: 663
Connection: close
Content-Type: text/html
```

tühi rida

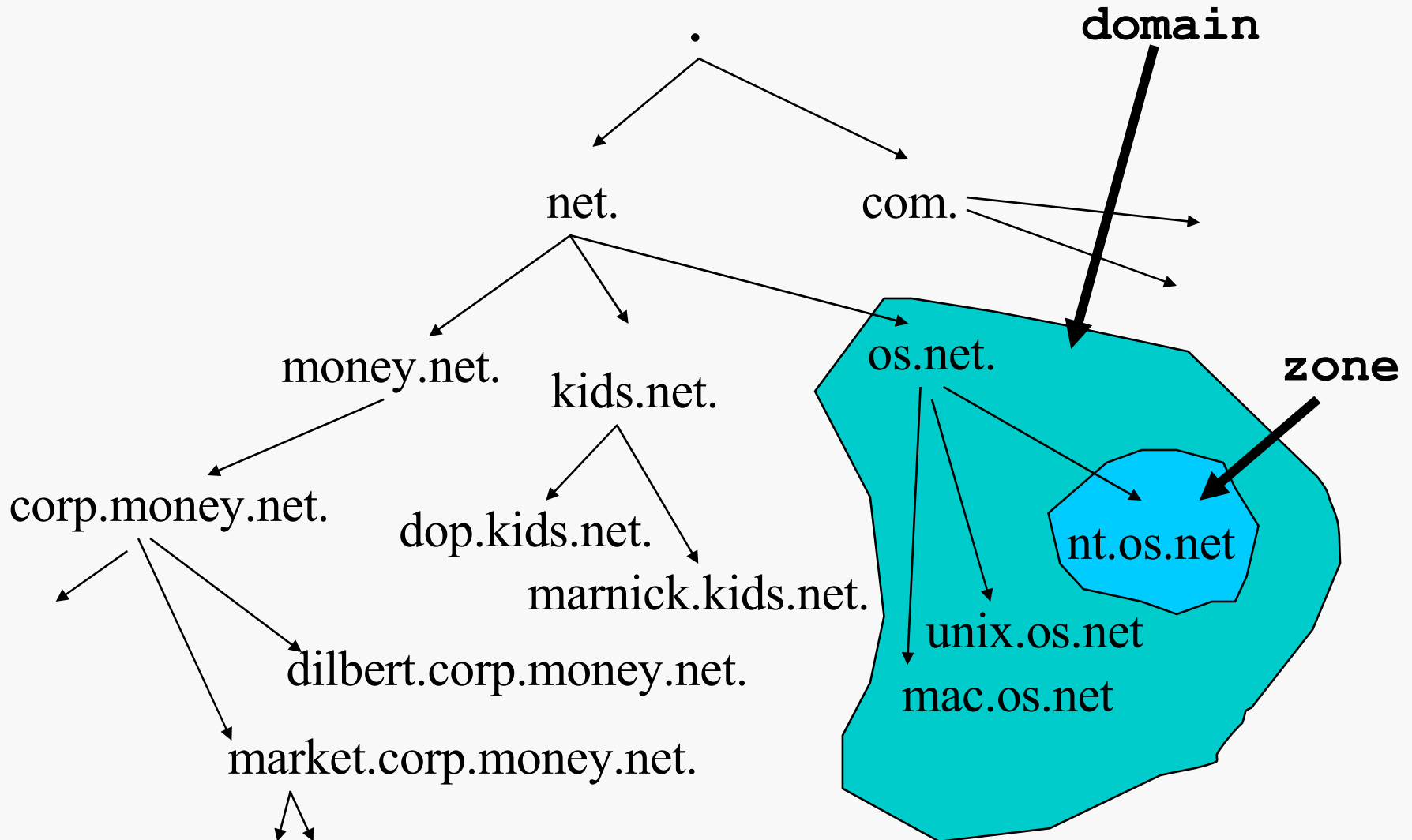
```
<html>
<head>
  <META HTTP-EQUIV="Content-type" CONTENT="text/html;
  charset=ISO-8859-1">
```

Tegelik
sisu

Edasi HTTP suunas: DNS: Domain Name Server

- **Ülesanne:** kui programmile on antud masina nimi, millega ühendust võtta (näiteks, www.cnn.com), siis tuleb kõigepealt leida sellenimelise masina IP aadress.
- **DNS serverid:** serverid, mis sisaldavad “nimi<->IP aadress” tabeleid ja vastavad päringutele “anna selle nime IP aadress”
- Kuna DNS päring tuleb teha iga hariliku www-päringu jaoks, siis neid päringuid tehakse maailmas tohutult palju (rohkem kui mistahes muid infopäringuid). Üks masin või kanal ei pea seda vastu.
- Lahendus: **hajutatud andmebaas**. DNS servereid on maailmas väga palju. Igaüks saab üles panna oma DNS serveri. DNS serverid süngivad omavahel infot oma tabelites.

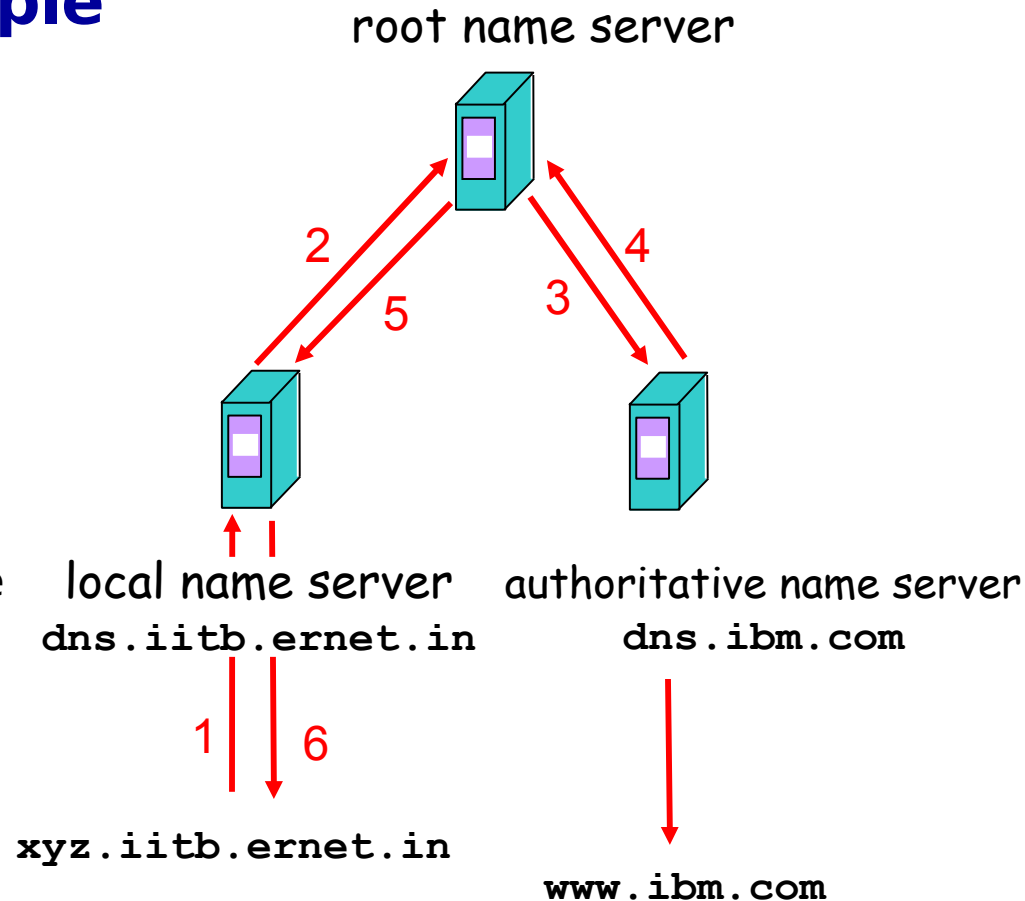
a DNS tree



DNS: Process example

- **Host xyz.iitb.ernet.in wants IP address of www.ibm.com**

3. Contacts its local DNS server, dns.iitb.ernet.in
4. dns.iitb.ernet.in contacts root name server, if necessary
5. root name server sends the authoritative name server address to local name server if necessary
- local name server asks the authoritative name server dns.ibm.com for the IP address of www.ibm.com and gets it
7. local name server sends the IP address to the host xyz.iitb.ernet.in



Kogu protsess HTTP näitel (klient)

■ HTTP

- Annab välja faile serverarvutist ja veidi lisainfi (HTTP päis)
- Server asub TCP pordil 80

% **telnet deepthought.ttu.ee 80**

- küsitakse nimeserverilt “deepthought.ttu.ee” IP aadress
 - nimeserver teeb DNS protokolliga rekursiivse otsingu
 - annab tulemuse 193.40.254.179
- telnet loob TCP ühenduse
 - reserveeritakse kohalik TCP port
 - saadetakse serverile TCP ühenduse avamise kutse

Kogu protsess HTTP näitel (võrk)

- telnet saadab päringu
 - saadetakse TCP paketti sisaldav IP datagramm 193.40.254.179 le
- marsruutimistabelist selgub, et tuleb kasutada vaikimisi marsruutimisreeglit (Näiteks: 192.168.0.1)
- ARP abil tehakse kindlaks selle Etherneti aadress
 - saadetakse üldlevi aadressil küsimus 192.168.0.1
 - saadakse vastus 00:E0:1E:67:F4:3F
- saadetakse Etherneti kaader aadressil 00:E0:1E:67:F4:3F
 - oodatakse kuni Etherneti võrk on vaba
 - saadetakse kaader
- 192.168.0.1-s ja teistes marsruuterites korratakse marsruutimisprotseduuri

Kogu protsess Http näitel (server)

- deepthoughti IP kiht annab datagrammi andmeosa edasi TCP kihile
- deepthoughti TCP kiht annab pordi numbri järgi andmeosa edasi http serveri-programmile
- http server-programm töötleb päringut
- http server-programm saadab vastuse ja lõpetab TCP ühenduse
- kordub protseduur läbi TCP, IP ja Etherneti

NetBIOS perekond

Rakenduskiht	SMB	Telnet Rlogin SMTP FTP HTTP jt	NFS DNS SNMP TFTP BOOTP jt
Esituskiht			
Seansi kiht	NetBIOS	TCP	UDP
Transpordi kiht	NetBEUI		
Võrgukiht		IP ICMP, IGMP	
Ühenduse kiht	ARP, RARP		SLIP, CSLIP PPP
Füüsiline kiht	Ethernet, IEEE 802.3	FDDI Token Ring ISDN jt	
	10 Base 2/F/T 100 Base-T		RS-323

NetBEUI

- **NetBIOS Extended User Interface**
- **Asendab kohtvõrgu liikluses TCP+IP**
 - + Optimeeritud väikeste kohtvõrkude jaoks
 - + Kiire
 - + Hea veakindlus
 - Pole marsruuditav
 - Palju liiklust üldaadressile

NetBIOS

- **NetBIOS (Network Basic Input/Output System)**
- **Võimaldab seansi ja datagrammi tüüpi ühendust**
- **Pakub nimeteenust**
 - Masina nimi kuni 15 tähte, hierarhiata
 - Meetodid nime teisendamiseks IP aadressiks:
 - küsib WINS serverilt (nimede ja IP aadresside andmebaas)
 - Küsimine üldlevi aadressil
 - LMHOSTS ja HOSTS fail
 - DNS päring